JC09 Rec'd PCT/PTO 28 FEB 2001

FORM PTO-1390 (REV 10-2000) U S DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER: T2146-906833

U.S. APPLICATION NO. (If known, see 37 CFR 1 5): 09/763868

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/FR00/01814	29 June 2000	30 June 1999

TITLE OF INVENTION: METHOD FOR MAKING SECURE A SENSITIVE INFORMATION PROCESSING IN A MONOLITHIC SECURITY MODULE, AND ASSOCIATED SECURITY MODULE

APPLICANT(S) FOR DO/EO/US: Michel HAZARD

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. [X] This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. [ ] This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. [X] This is an express request to promptly begin national examination procedures (35 U.S.C. 371(f)).
4. [X] The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
5. [X] A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. [ ] is attached hereto (required only if not communicated by the International Bureau).
   b. [X] has been communicated by the International Bureau.
   c. [ ] is not required, as the application was filed in the United States Receiving Office (RO/US).
6. [X] An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
7. [ ] Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. [ ] are attached hereto (required only if not communicated by the International Bureau).
   b. [ ] have been communicated by the International Bureau.
   c. [ ] have not been made; however, the time limit for making such amendments has NOT expired.
   d. [ ] have not been made and will not be made.
8. [ ] An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. [X] An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. [ ] An English language translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11 to 16 below concern document(s) or information included:

11. [X] An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. [X] An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
    to BULL CP8
13. [X] A FIRST preliminary amendment.
    [ ] A SECOND or SUBSEQUENT preliminary amendment.
14. [ ] A substitute specification.
15. [X] A change of power of attorney and/or address letter.
16. [X] Other items or information:
    Verification of translation
    Proposed Drawing Corrections & 7 red-lined formal drawings
    Copies of PCT forms: PCT/IB/ 301 & 308, PCT/RO/101, first page of published International Application

U.S. APPLICATION NO (if known, see 37 CFR 1.5)	INTERNATIONAL APPLICATION NO	ATTORNEY'S DOCKET NUMBER
09/763868	PCT/FR00/01814	T2146-906833

17. [X] The following fees are submitted:

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :

	CALCULATIONS PTO USE ONLY
Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO **$1000.00**	
International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO **$860.00**	
International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO **$710.00**	
International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) **$690.00**	
International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) **$100.00**	
ENTER APPROPRIATE BASIC FEE AMOUNT =	$ 860.00
Surcharge of **$130.00** for furnishing the oath or declaration later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(e)).	$

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE	
Total claims	19 - 20 =	0	X $18.00	$
Independent claims	2 - 3 =	0	X $80.00	$
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ $270.00	$
TOTAL OF ABOVE CALCULATIONS =				$
[ ] Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above are reduced by 1/2.				$ 860.00
SUBTOTAL =				$ 860.00
Processing fee of **$130.00** for furnishing the English translation later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(f)). +				$
TOTAL NATIONAL FEE =				$
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00** per property +				$ 40.00
TOTAL FEES ENCLOSED =				$ 900.00
			Amount to be refunded:	$
			charged:	$

a. [X] A check in the amount of $ 900.00 to cover the above fees is enclosed.

b. [ ] Please charge my Deposit Account No. _____ in the amount of $_____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. [X] The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 501165. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO.

Edward J. Kondracki
MILES & STOCKBRIDGE P.C.
1751 Pinnacle Drive, Suite 500
McLean, VA 22102-3833

Edward J. Kondracki
SIGNATURE:

Edward J. Kondracki
NAME

20,604
REGISTRATION NUMBER

09/763868
JC02 Rec'd PCT/PTO 28 FEB 2001

IN THE UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)

Applicant: Michel HAZARD

International Application No.: PCT/FR00/01814

International Filing Date: 29 June 2000

U.S. Serial No.: To be Assigned

U.S. Filing Date: February 28, 2001

For: **METHOD FOR MAKING SECURE A SENSITIVE INFORMATION PROCESSING IN A MONOLITHIC SECURITY MODULE, AND ASSOCIATED SECURITY MODULE**

McLean, Virginia

PRELIMINARY AMENDMENT

Honorable Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Sir:

Please amend the subject application, filed concurrently herewith, as indicated below:

IN THE TITLE:

After "METHOD FOR", please delete "MAKING SECURE A SENSITIVE INFORMATION PROCESSING" and substitute –PROTECTING THE PROCESSING OF SENSITIVE INFORMATION--;

IN THE SPECIFICATION:

After the title and before the first paragraph on page 1, insert the following heading at the left-hand margin:

--FIELD OF THE INVENTION--;

Page 1, at line 14, and before the paragraph beginning "The term monolithic...", insert the following heading at the left-hand margin:

--BACKGROUND OF THE INVENTION--;

Page 2, at line 21, and before the paragraph beginning "The object of the ..." insert the following paragraph at the left-hand margin:

--SUMMARY OF THE INVENTION--;

Page 3, at line 13 and before the paragraph beginning "Other characteristics...", insert the following heading at the left hand margin:

--BRIEF DESCRIPTION OF THE DRAWINGS--;

Page 3, at line 27 and before the paragraph beginning "The wiring diagram...", insert the following heading at the left hand margin:

--DESCRIPTION OF THE PREFERRED EMBODIMENT(S)--;

Page 14, after the final paragraph, insert the following new paragraph:

--While this invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the preferred embodiments of the invention as set forth herein, are intended to be illustrative, not limiting. Various changes may be made without departing from the true spirit and full scope of the invention as set forth herein and defined in the claims.—

IN THE CLAIMS:

Please cancel claims 1 – 19 in their entirety and without prejudice and substitute the following new claims:

--20. A method for protecting the processing of sensitive information in a security module having a monolithic structure, information processing means (31) and storage means for storing (32,33) information capable of being processed by said processing means, comprising the following steps:

- selecting a piece of sensitive information stored in the storage means;
- determining a specific condition for the integrity of said information;
- reading the information and transmitting it to the processing means for processing;
- processing the information and verifying during processing that the specific condition is satisfied; and
- disabling processing of the information if the specific condition is not satisfied.

21. The method according to claim 20, wherein the information is an operation code read in the storage means (32, 33), the operation code being contained in a table having a content determined during the manufacture of the security module, and the specific condition for the integrity of the information being the value of the information is equal to one of several set values.

22. The method according to claim 21, wherein the operation code to be processed is coded in the form of data bits and said bits do not all have the same binary value.

23. The method according to claim 20, wherein the specific step of determining the condition for the integrity of said information comprises checking a calculated or first piece of integrity data using the information read in the storage means (32, 33) during the reading of the information and transmitting the first piece of integrity data to the processing means, and calculating a second piece of integrity data by the processing means from the information received and checking for equality between the first and second pieces of integrity data.

24. The method according to claim 23, wherein the first piece of integrity data is calculated from at least one piece of calculation data whose value varies as a function of time.

25. The method according to claim 23, wherein the first piece of integrity data is calculated from at least one piece of calculation data whose value varies randomly.

26. The method according to claim 20, wherein the disabling of the processing of the information is performed by a microprogrammed instruction.

27. The method according to claim 26, wherein the microprogrammed instruction performs the following steps:

- writing a piece of disable data into a nonvolatile location of the storage means (32, 33); and
- disabling the processing of the information.

28. The method according to claim 27 further comprising reading by the processing means (31) a nonvolatile location of the storage means (32, 33) upon power up of said module and disabling the module if a value read at this location does not match.

29. A security module comprising an electronic circuit having a monolithic structure and comprising information processing means (31) and information storage means (32, 33), means for extracting information from the storage means and means for selecting information to be processed, the processing means further comprising means for verifying a specific integrity condition of a piece of sensitive information, and means for disabling the processing of the information, said means for disabling being activated when the means for verification have detected that the specific condition is not satisfied.

30. A security module according to claim 29, wherein the processing means (31) execute instructions corresponding to operation codes extracted from a table, characterized in that the table comprises a forbidden instruction value.

31. A security module according to claim 30, wherein the operation code to be processed is coded in the form of data bits, the security module comprising a means for reading the values of all the bits and a disabling means activated when the values of the bits are all identical.

32. A security module according to claim 29, wherein the processing means (31) execute instructions corresponding to operation codes extracted from a table, the security module comprising a means for reading an operation code and a disabling means activated during the reading of a forbidden operation code.

33. A security module according to claim 32, wherein the disabling means comprise a means for irreversibly writing an indicator into the storage means (32, 33), and a means for reading said indicator during the next power-up of the module.

34. A security module according to claim 29, comprising parity generators (7, 8) cooperating with the storage means, parity generators (11) cooperating with the processing means, and a comparator connected to each of the parity generators and capable of inducing an interrupt in the processing means.

35. A security module according to claim 34, wherein the operation of the parity generators (7, 8) varies as a function of time.

36. A security module according to claim 34, wherein the operation of

the parity generators (7, 8) varies randomly.

37 A security module according to claim 33, characterized in that the irreversible writing of the indicator into the storage means (32, 33) is performed by executing a microprogrammed instruction.

38. A security module according to claim 29, characterized in that the security module is a microcircuit card.--

EFS: 2006-12-05

IN THE ABSTRACT:

Please cancel the Abstract at page 18 and substitute the following Abstract.

Express Mail Label No. EV 339 908 925 US

-- **ABSTRACT**

The invention relates to a method and associated security module for protecting the processing of sensitive information in a security module with a monolithic structure, the module comprising information processing means (9) and means for storing (3, 4) information capable of being processed by said processing means. The method comprises the following steps:

- selecting a piece of sensitive information in the storage means;
- determining (7) a specific condition for the integrity of said information;
- reading the information and transmitting (1) it to the processing means;
- verifying (11) during the processing of the information that the specific condition is satisfied; and
- disabling the processing of the information if the specific condition is not satisfied.--

REMARKS

This Preliminary Amendment is filed to insert headings to conform the application to U.S. practice and to correct informalities in the specification, claims and abstract resulting from a literal translation of the French text.

Early action on the merits is earnestly solicited.

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: February 28, 2001

By: _____
Edward J. Kondracki
Registration No. 20,604

1751 Pinnacle Drive – Suite 500
McLean, VA 22102-3833
Tel.: 703/903-9000
Fax: 703/610-8686

09/763868
JC02 Rec'd PCT/PTO 28 FEB 2001

T2146-906833-US3830/BC(PCT)

IN THE UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)

Applicant: Michel HAZARD

International Application No.: PCT/FR00/01814

International Filing Date: 29 June 2000

U.S. Serial No.: To be Assigned

U.S. Filing Date: February 28, 2001

For: **METHOD FOR MAKING SECURE A SENSITIVE INFORMATION PROCESSING IN A MONOLITHIC SECURITY MODULE, AND ASSOCIATED SECURITY MODULE**

McLean, Virginia

PROPOSED DRAWING CORRECTIONS

Hon. Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

Applicant requests approval of the drawing corrections on Figs. 1 – 7 as shown in red on the attached seven (7) sheets.

The proposed corrections only comprise translating the French terms into English and removing the headings "1/7" – "7/7" to conform the drawings to U.S. practice.

09/763868
JC02 Rec'd PCT/PTO 28 FEB 2001

T2146-906833-US3830/BC(PCT)

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: February 28, 2001

By: Edward J. Kondracki
Edward J. Kondracki
Registration No. 20,604

1751 Pinnacle Drive – Suite 500
McLean, VA 22102-3833
Tel.: 703/903-9000
Fax: 703/610-8686

PCT/US00/05656

091763868

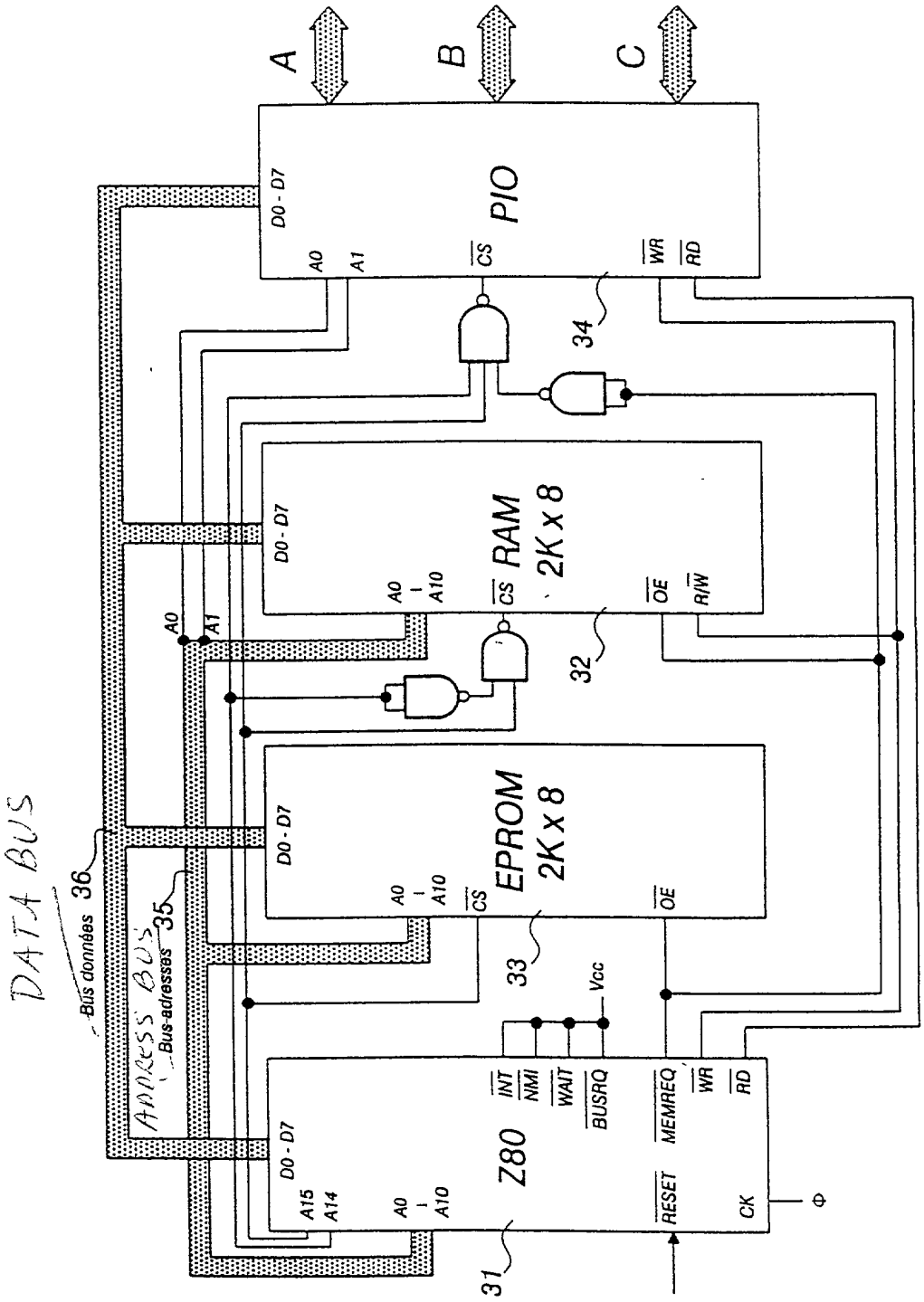

Fig.1

HIGH-ORDER

\Poids-forts / LOW-ORDER \Poids-faibles	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	BRSET0	BSET0	BRA	NEG	NEG	NEG	NEG	NEG	RTI		SUB	SUB	SUB	SUB	SUB	SUB
1 0001	BRSET1	BSET1	BRN						RTS		CMP	CMP	CMP	CMP	CMP	CMP
2 0010	BRSET2	BSET2	BHI								SBC	SBC	SBC	SBC	SBC	SBC
3 0011	BRSET3	BSET3	BLS	COM	COM	COM	COM	COM	SWI		CPX	CPX	CPX	CPX	CPX	CPX
4 0101	BRSET4	BSET4	BCC	LSR	LSR	LSR	LSR	LSR			AND	AND	AND	AND	AND	AND
5 0101	BRSET5	BSET5	BCS								BIT	BIT	BIT	BIT	BIT	BIT
6 0110	BRSET6	BSET6	BNE	ROR	ROR	ROR	ROR	ROR			LDA	LDA	LDA	LDA	LDA	LDA
7 0111	BRSET7	BSET7	BEQ	ASR	ASR	ASR	ASR	ASR		TAX		STA	STA	STA	STA	STA
8 1000	BRCLR0	BCLR0	BHCC	LSL	LSL	LSL	LSL	LSL		CLC	EOR	EOR	EOR	EOR	EOR	EOR
9 1001	BRCLR1	BCLR1	BHCS	ROL	ROL	ROL	ROL	ROL		SEC	ADC	ADC	ADC	ADC	ADC	ADC
A 1010	BRCLR2	BCLR2	BPL	DEC	DEC	DEC	DEC	DEC		CLI	ORA	ORA	ORA	ORA	ORA	ORA
B 1011	BRCLR3	BCLR3	BMI							SEI	ADD	ADD	ADD	ADD	ADD	ADD
C 1100	BRCLR4	BCLR4	BMC	INC	INC	INC	INC	INC		RSP		JMP	JMP	JMP	JMP	JMP
D 1101	BRCLR5	BCLR5	BMS	TST	TST	TST	TST	TST		NOP	BSR	JSR	JSR	JSR	JSR	JSR
E 1110	BRCLR6	BCLR6	BIL								LDX	LDX	LDX	LDX	LDX	LDX
F 1111	BRCLR7	BCLR7	BIH	CLR	CLR	CLR	CLR	CLR	WAIT	TXA		STX	STX	STX	STX	STX

Fig.2

HIGH-ORDER

\Poids-forts \Poids-faibles LOW-ORDER	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	FRD	RTI	BSET0	BSET0	BRA	NEG	NEG	NEG	NEG	NEG	SUB	SUB	SUB	SUB	SUB	SUB
1 0001	FRD	RTS	BSET1	BSET1	BRN	FRD	FRD	FRD	FRD	FRD	CMP	CMP	CMP	CMP	CMP	CMP
2 0010	FRD	FRD	BSET2	BSET2	BHI	FRD	FRD	FRD	FRD	FRD	SBC	SBC	SBC	SBC	SBC	SBC
3 0011	FRD	SWI	BSET3	BSET3	BLS	COM	COM	COM	COM	COM	CPX	CPX	CPX	CPX	CPX	CPX
4 0101	FRD	FRD	BSET4	BSET4	BCC	LSR	LSR	LSR	LSR	LSR	AND	AND	AND	AND	AND	AND
5 0101	FRD	FRD	BSET5	BSET5	BCS	FRD	FRD	FRD	FRD	FRD	BIT	BIT	BIT	BIT	BIT	BIT
6 0110	FRD	FRD	BSET6	BSET6	BNE	ROR	ROR	ROR	ROR	ROR	LDA	LDA	LDA	LDA	LDA	LDA
7 0111	TAX	FRD	BSET7	BSET7	BEQ	ASR	ASR	ASR	ASR	ASR	STA	STA	STA	STA	STA	FRD
8 1000	CLC	FRD	BCLR0	BCLR0	BHCC	LSL	LSL	LSL	LSL	LSL	EOR	EOR	EOR	EOR	EOR	EOR
9 1001	SEC	FRD	BCLR1	BCLR1	BHCS	ROL	ROL	ROL	ROL	ROL	ADC	ADC	ADC	ADC	ADC	ADC
A 1010	CLI	FRD	BCLR2	BCLR2	BPL	DEC	DEC	DEC	DEC	DEC	ORA	ORA	ORA	ORA	ORA	ORA
B 1011	SEI	FRD	BCLR3	BCLR3	BMI	FRD	FRD	FRD	FRD	FRD	ADD	ADD	ADD	ADD	ADD	ADD
C 1100	RSP	FRD	BCLR4	BCLR4	BMC	INC	INC	INC	INC	INC	JMP	JMP	JMP	JMP	JMP	FRD
D 1101	NOP	FRD	BCLR5	BCLR5	BMS	TST	TST	TST	TST	TST	JSR	JSR	JSR	JSR	JSR	BSR
E 1110	FRD	FRD	BCLR6	BCLR6	BIL	FRD	FRD	FRD	FRD	FRD	LDX	LDX	LDX	LDX	LDX	LDX
F 1111	TXA	WAIT	BCLR7	BCLR7	BIH	CLR	CLR	CLR	CLR	CLR	STX	STX	STX	STX	STX	FRD

Fig.3

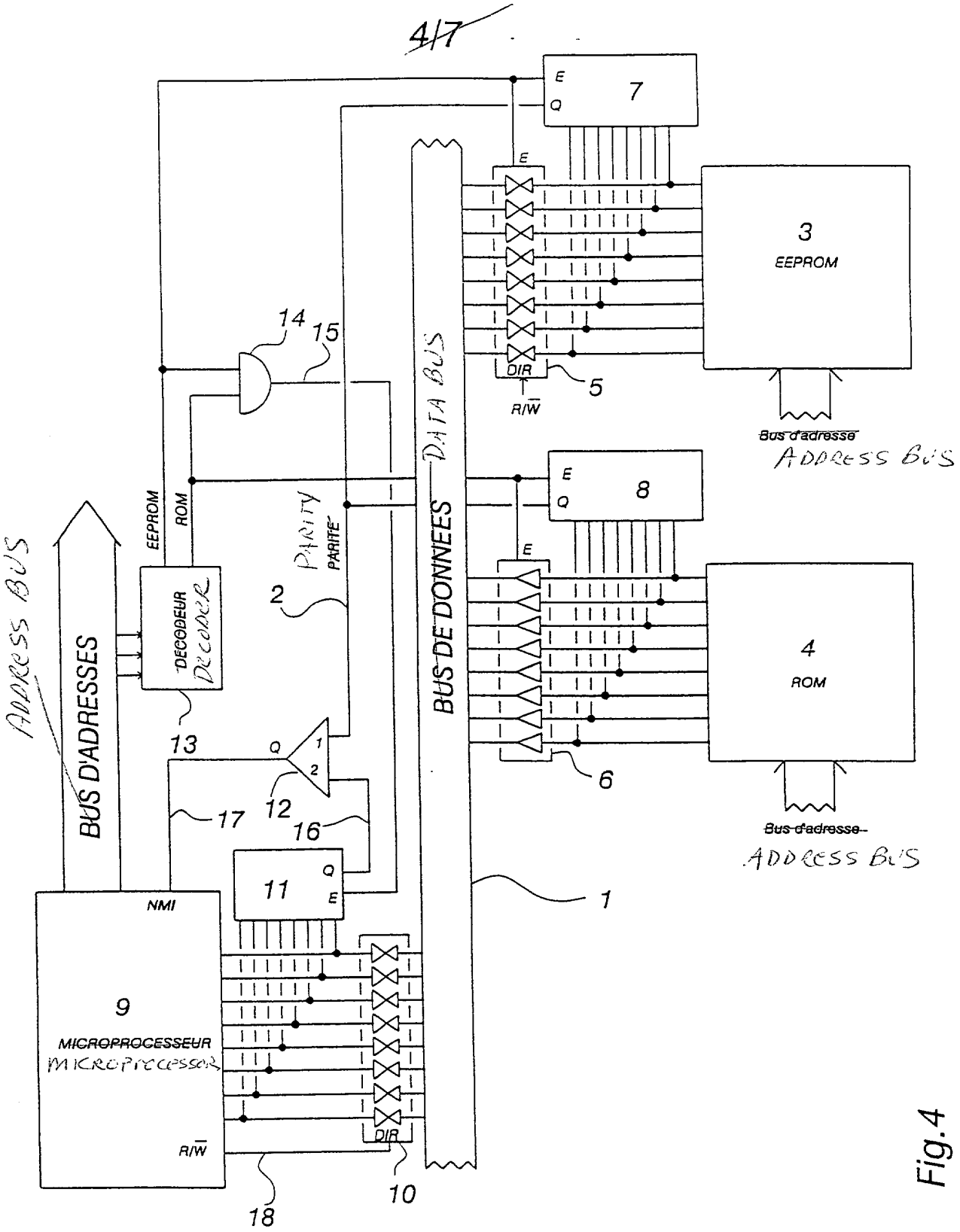
E
Q
7
E
3
EEPROM
DIR
R/W
5
Bus d'adresse
ADDRESS BUS
14
15
DATA BUS
BUS DE DONNEES
E
Q
8
E
4
ROM
6
Bus d'adresse
ADDRESS BUS
EEPROM
ROM
PARITY
PARITE
2
DECODEUR
Decoder
13
Q
1
2
12
17
16
11
Q
E
1
ADDRESS BUS
BUS D'ADRESSES
NMI
9
MICROPROCESSEUR
MICROPROCESSOR
R/W
DIR
10
18

Fig.4

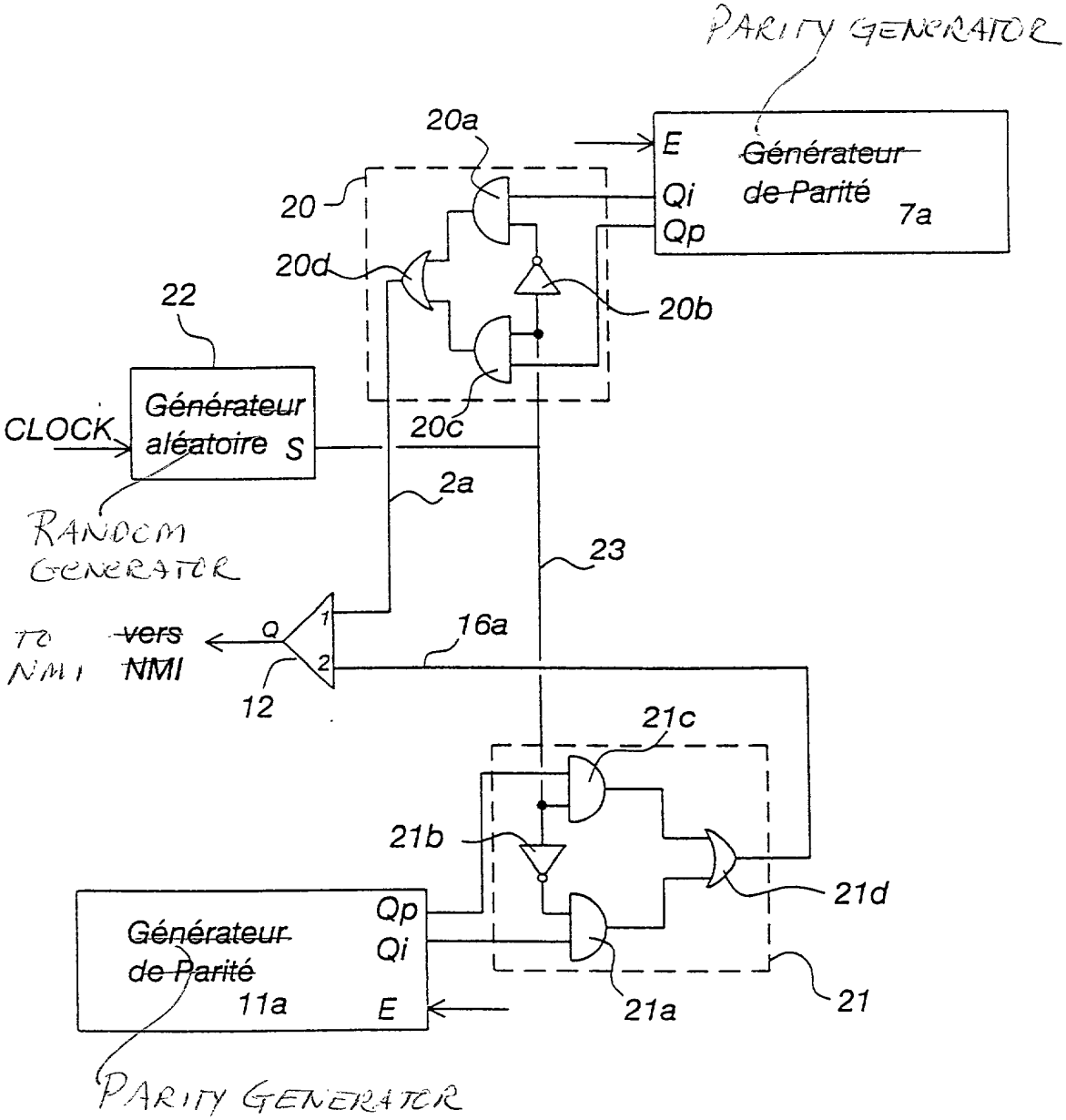
PARITY GENERATOR
20a
E
Générateur
de Parité
7a
Qi
Qp
20
20d
20b
22
Générateur
aléatoire
S
CLOCK
20c
2a
RANDOM
GENERATOR
23
TO
NMI
vers
NMI
Q
1
2
16a
12
21c
21b
21d
Générateur
de Parité
11a
Qp
Qi
E
21a
21
PARITY GENERATOR

Fig.5

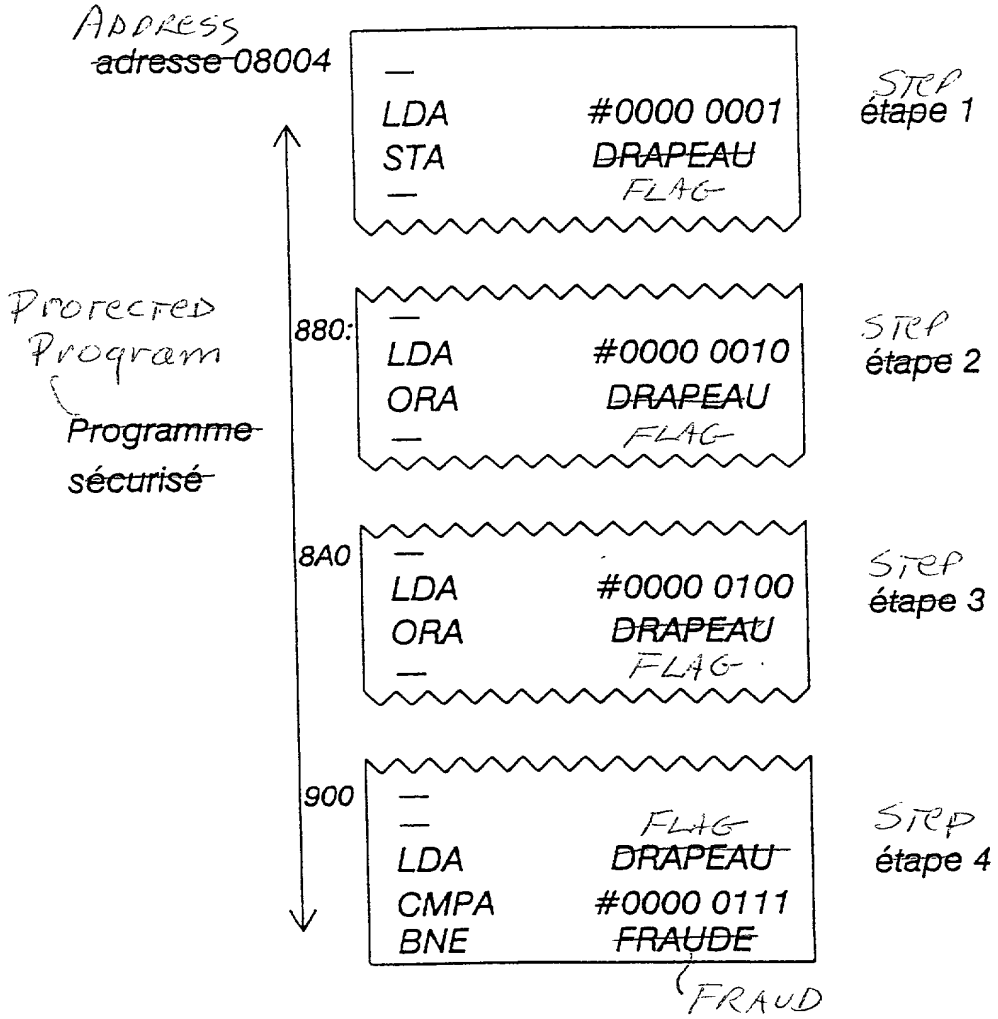
ADDRESS
adresse 08004
Protected Program
Programme sécurisé
—
LDA #0000 0001
STA DRAPEAU
— FLAG
STEP
étape 1
880:
—
LDA #0000 0010
ORA DRAPEAU
— FLAG
STEP
étape 2
8A0
—
LDA #0000 0100
ORA DRAPEAU
— FLAG
STEP
étape 3
900
—
— FLAG
LDA DRAPEAU
CMPA #0000 0111
BNE FRAUDE
FRAUD
STEP
étape 4

Fig.6

7/7

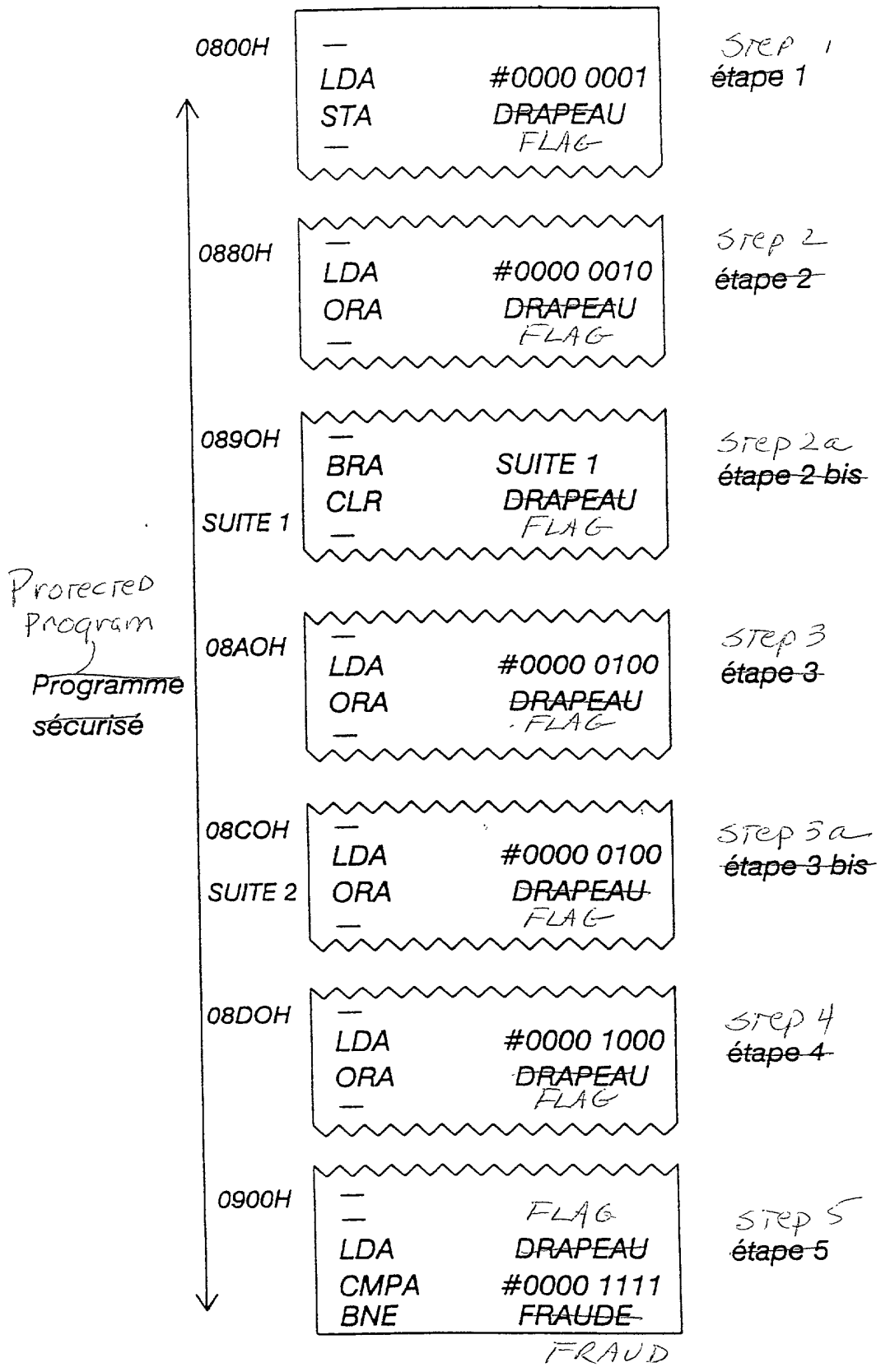
0800H
—
LDA #0000 0001
STA DRAPEAU
— FLAG
Step 1
étape 1
0880H
—
LDA #0000 0010
ORA DRAPEAU
— FLAG
Step 2
étape 2
089OH
—
BRA SUITE 1
CLR DRAPEAU
SUITE 1 — FLAG
Step 2a
étape 2 bis
Protected Program
Programme sécurisé
08AOH
—
LDA #0000 0100
ORA DRAPEAU
— FLAG
Step 3
étape 3
08COH
—
LDA #0000 0100
SUITE 2 ORA DRAPEAU
— FLAG
Step 3a
étape 3 bis
08DOH
—
LDA #0000 1000
ORA DRAPEAU
— FLAG
Step 4
étape 4
0900H
—
— FLAG
LDA DRAPEAU
CMPA #0000 1111
BNE FRAUDE
FRAUD
Step 5
étape 5

Fig.7

09/763868
JC02 Rec'd PCT/PTO 2 8 FEB 2001
7/PRTS

METHOD FOR MAKING SECURE A SENSITVE INFORMATION PROCESSING IN A MONOLITHIC SECURITY MODULE, AND ASSOCIATED SECURITY MODULE

The present invention relates to a method and a device for increasing the security of a monolithic security module comprising a microprocessor and designed to execute a program to be protected. A program to be protected is a set of instructions whose execution must meet criteria such as: authentication of the user, confidentiality of the data exchanged, authenticity and validity of a transaction, and generally, application data processing in which a user's rights and obligations are controlled. The object of the present invention is to provide the security module with means for detecting and counteracting fraudulent external interventions for accessing sensitive data by circumventing intrinsically known prior controls.

The term monolithic "security module" covers any electronic circuit constituted by a single silicon chip and encased in a container, the circuit comprising a central processor, at least one memory, and peripherals such as input/output circuits, interrupt controllers, etc. The security module can take the form of an integrated circuit or a portable object of the chip-card type used, for example, in the banking field, in mobile radiotelephones, pay television decoders, health, transportation, etc.

The security modules according to the invention include at least one microprocessor, a memory containing a program and input/output means for communicating with the outside. The memory contains at least one application program that can be written into a memory of the ROM type during the manufacture of the circuit or subsequently written into a programmable memory. Generally, the program contains the instructions executed by the microprocessor, the instructions being transferred from the ROM or nonvolatile memory to the microprocessor through a data bus coupled with an address bus. If the bus paths are too long, the manufacturer of the circuit places amplifier circuits at the ends of the buses so that the binary signal is correctly propagated along its entire length.

Amplifier circuits require high input impedance, which makes them susceptible to external interference. When subjected to elementary particle radiation, they can be

saturated for a certain amount of time, and no matter what the incoming data, emit only 0 volt or +Vcc through the bus, i.e. a binary "0" or "1." This radiation may be constituted by alpha rays, X-rays, or positively or negatively charged ions depending on whether an output of "0" or "1" from the amplifier is desired.

A hacker in possession of an authentic card may try to evade the security devices in order to misuse services: he will then subject the security module to such radiation, hoping to interfere with its operation to his advantage. The emission source can be short enough to interfere with the execution of only one or several instructions. This interference can replace the value read in the memory of the program with another value imposed by the emission of the radiation. Thus, the program flow is different and the hacker can use this to his advantage.

A first way to counteract this type of attack is to install radiation sensors; as soon as these sensors detect any abnormal radiation, they set a flag that can be read by the program. Such a solution is described in US patent 5.465.349, the American equivalent of French patent 2 688 274. But hackers have improved the means of investigation and are using emitters of extremely fine radiation. By aiming the radiation only at the amplifiers or at a group of amplifiers, the sensors no longer detect the fraud. No matter what, this counterattack is incapable of detecting an intervention by means of micropoints, and in any case it is possible to disturb the sensors' read sequence so that the program runs as though nothing abnormal has been detected.

The object of the invention is to detect alterations in the transmission of the instructions and the data between a memory and a microprocessor. The problem that the invention seeks to solve is the detection of an external intervention attempting to interfere with the flow of a program by modifying the value of the signals exchanged through the busses of a microcontroller.

This problem is solved according to the invention by providing for measurements that allow the security modules to verify whether the information is flowing properly through the module and whether the programs have been fully executed. If not, the normal execution of the program is interrupted. Optionally, any further utilization of the module is prohibited, because the module has detected that it is being used illicitly.

More precisely, the invention claims a method for protecting the processing of sensitive information in a security module with a monolithic structure, the module comprising information processing means and means for storing information capable of being processed by said processing means, characterized in that it comprises the following steps:

- selecting a piece of sensitive information in the storage means;
- determining a specific condition for the integrity of said information;
- reading the information and transmitting it to the processing means;
- verifying during the processing of the information that the specific condition is satisfied;
- disabling the processing of the information if the specific condition is not satisfied.

Other characteristics and advantages of the present invention will emerge during the following description of several preferred but non-limiting embodiments, in reference to the attached drawings, in which:

Fig. 1 represents the diagram of a conventional microprocessor system.

Fig. 2 shows the instruction set of the MOTOROLA 6805 processor.

Fig. 3 shows the modified instruction set of the microprocessor according to the invention.

Fig. 4 shows a block diagram of the modified circuit according to the invention.

Fig. 5 shows a part of the block diagram of the modified circuit according to an improvement of the invention.

Fig. 6 shows a part of the program memory, equipped with steps for updating and modifying an indicator that makes it possible to detect external interference.

Fig. 7 shows a part of the program memory according to a variant of the invention.

The wiring diagram of a microprocessor system is represented in Fig. 1. This diagram is extracted from the book *Architecture de l'Ordinateur* (*Computer Architecture*) by Andrew Tanenbaum, published by InterEditions. The system comprises, in an intrinsically known way, the following elements: a microprocessor 31, a RAM 32, a program memory EEPROM 33 containing the executable program, and input/output

means 34. The these various elements are connected by two busses, the address bus 35 and the data bus 36. It is customary to refer to the bus that carries the set of signals such as the clock signals, the read and write signals, etc., as the "control bus."

Depending on the operation executed, these elements can be in information receiving mode or information sending mode. When the microprocessor reads an instruction in a byte at a certain address, the eight cells of the ROM constituting the byte corresponding to this address send the value written in the cells through the data bus, which transmits them to the microprocessor, the latter being in receiving mode. Conversely, when the microprocessor sends a piece of data to the memory, the data bus is in the sending mode. The internal operation of the component is detailed in US patent 4.382.279, the American equivalent of French patent 2 461 301.

Each microprocessor has a specific instruction set. The MOTOROLA 6805 microprocessor processes eight-bit data internally. Its data bus comprises 8 lines and the operation code of its instructions is coded into 8 bits. For example, the instruction set of the M6805 and M1468805 microprocessors, extracted from the documentation "Microcomputer/Microprocessor User's Manual" is represented in Fig. 2. An instruction for this type of microprocessor is coded into 8 bits, hence there are 256 different codes, but not all of them are executable by the microprocessor. Certain codes are not implemented and correspond to holes in the table of Fig. 2, for example the codes: 90H, 31H, 41H, 51H, 61H, 71H, 91H, 23H, 24H, 25H, 26H, 27H, 28H 29H, 93H, 94H, etc. If the microprocessor reads an operation code that is not implemented, its execution is not guaranteed. Thus, the microprocessor can jump to the next instruction without doing anything else.

As may be seen in Fig. 2, the code "00" of the instruction set of the MOTOROLA 6805 microprocessor corresponds to the instruction "BRSET0" and the code "FF" corresponds to the indexed instruction "STX". The instruction "BRSET0" takes up three bytes in the program memory and is executed in ten clock cycles; the instruction "STX" takes up one byte and is executed in five clock cycles. The microprocessor has a ordinal counter containing the address of the instruction in the program memory that is currently being executed. As stated in the preamble, if a hacker sends radiation through the bus, one or another of these instructions will be executed instead of the one actually read in

the program memory. In the case of the instruction "BRSET0", after its execution, the ordinal counter of the microprocessor is increased by three units and hence points three bytes further ahead in the program memory. For the instruction "STX", the ordinal counter is increased by one unit. By emitting the radiation during the ten clock strokes required for the execution of BRSET0, the microprocessor will read three bytes at "00", and translate this into "test whether the bit 0 of the byte 00 is at 1 and jump if that is the case". Because of the radiation, the value 00 of the byte at the address 00 is read, so the jump does not happen and the program continues with the next instruction. Thus, by subjecting the circuit to radiation for a certain number of clock pulses, a hacker can artificially advance the ordinal counter in successive three-byte jumps, and prevent the execution of a sequence written in program memory. For a microprocessor from the 8051 family produced by the INTEL company and various other companies, the operation code whose binary value is 00 corresponds to the instruction NOP, which means "no operation." This instruction uses only one byte. In this case, the hacker can artificially advance the ordinal counter byte by byte.

A first solution consists of interrupting the normal operation of the microprocessor during the reading of an instruction whose code is "00" or "FF". The module becomes mute; just one power-down followed by one power-up of the module can restart the microprocessor. Fig. 3 shows the modified instruction set according to the invention. The new instruction corresponding to "00" or "FF" is called "FRAUD", abbreviated FRD. Compared to the normal instruction set described in Fig. 2, some of the columns have been inverted so that the codes "00" and "FF" no longer correspond to executable instructions. Column number "0", which contained the instruction "BRSET0" is transferred to column 2; column number "F" which contained the instruction "STX" is transferred to column A. The columns are transferred by modifying the mask of the circuit, in the element that decodes the four most significant bits of the operation code of the instructions. It would also be possible to modify the decoding of the four least significant bits of the operation code, but it wasn't necessary for this microprocessor model.

The program does not contain a "FRAUD" instruction. The progammer who writes an application program is careful not to implement such instructions in his program.

As shown in Fig. 3, the solution can be generalized to all of the operation codes that do not correspond to an instruction recognizable by the microprocessor. Thus, the new "FRAUD" instruction is assigned to all the codes not used, such as: 01H, 02H, 03H, 04H, 05H, 06H, 0EH, 12H, 14H, 15H, 16H, 17H, 18H, 19H, 1AH, and so on up to FFH

- 47 different operation codes in all, all of which execute the same "FRAUD" instruction.

In a variant, this interruption triggers an instruction microprogrammed into the microprocessor. This type of instruction executes a non-reversible instruction of the type for writing a code into nonvolatile memory. When the next power-up occurs, the reset management circuit tests the value of this code in nonvolatile memory and disables the operation of the microprocessor. The microprogrammed instruction triggers an operation for definitively disabling the circuit. Microprogrammed instructions have the advantage of being resident in the heart of the microprocessor and hence their execution cannot be interrupted or altered by radiation acting on the bus. It is therefore impossible to detect the execution of a microprogrammed disable instruction.

A second solution for detecting the disturbance of a piece of data in a bus is to implement an integrity check from one end of the busses to the other. Fig. 4 shows the wiring diagram with its bus integrity controller.

Added to the eight lines of the data bus 1 is a ninth line, marked PARITY 2, whose logical state corresponds to the parity value calculated from the eight pieces of binary information present in the data bus. As stated above, the operation codes of the various instructions of a program are read from the ROM 3 or from nonvolatile programmable memory 4, EEPROM for example. The signal output from a memory cell, whose value represents the binary data stored by this cell, is too weak to reach the microprocessor via the data bus. In order for these signals to reach the microprocessor, amplifier circuits 5 and 6 are installed just after the cells of the memories EEPROM and ROM and before the bus. These two circuits have a control input E that makes it possible to activate their output. If the signal E has a level "1," the eight outputs are not active.

They are in a state of high impedance. The EEPROM being read-and write-accessible, the amplifier circuit 5 is two-way. The direction of the data transfers is controlled by the microprocessor through the READ/WRITE control line 18.

The means for checking the integrity of the data read in the memory cells comprise parity generators 7, 8 and 11, a comparator 12 and a non-maskable interrupt input called "NMI" connected to the microprocessor. The parity generators 7, 8 and 11 calculate the parity value of the data present in eight inputs. For example, the circuit 74HC280 manufactured by National Semiconductor is a parity generator available in the form of an integrated circuit. Its structure can be fully integrated into the monolithic circuit. The parity generators also have an output line Q whose state represents the parity value of the data applied to the eight inputs and a control line E that activates the output Q when 0 volt is applied to it. In this case, the output line Q is at 0 volt if the number of inputs at +Vcc is even, or at +Vcc if the number of intputs at +Vcc is odd. When +Vcc is applied to the input E, the output line Q is in a state of high impedance. The parity generator 7 calculates the parity of the data selected in the EEPROM, the parity generator 8 calculates the parity of the data selected in the ROM and the parity generator 11 calculates that present in the microprocessor.

The microprocessor 9 also has an amplifier circuit 10 for ensuring the compatibility of the signals flowing through the data, address and control busses. This amplifier circuit connected to the data bus 10 is two-way; on one end, it amplifies the data signals that the microprocessor sends to the memories and to the peripherals; on the other end it amplifies the signals received by these same memories and peripherals so that the signals are correctly processed by the internal circuits of the microprocessor. Most microprocessors have a "fetch" device that makes it possible, in the same clock cycle, to execute an instruction and to read the operation code of the next instruction.

The microprocessor 9 controls the address bus; the value applied to this bus determines which type of peripheral is selected. In order to reduce the number of selection lines, an address decoder 13 receives the address bus in an input and selects the various peripherals and memories through specialized lines: the line called "EEPROM" selects the EEPROM and the line "ROM" selects the ROM. Each of these two lines is connected to the memory and to the corresponding parity generator.

The lines EEPROM and ROM are active in the 0 state. The output of the AND gate 14 generates a selection signal common to both memories; if one of the two lines EEPROM and ROM is at "0", the line 15 is also at "0". The line 15 is connected to the input E of the parity generator 11 which, as a result, is active at the same time as one of the parity generators 7 or 8. If no memory is selected, the output lines 2 and 16 of the generators are in the high impedance state. The output lines 2 and 16 of the generators are respectively connected to each of the two inputs of the comparator 12. The output Q of the comparator is connected to an interrupt input of the microprocessor (marked "NMI" in Fig. 4).

When in use, the microprocessor reads an operation code representing an instruction in the ROM. It applies the address of the code to the address bus, which activates the line ROM, and consequently the amplifiers 6 and the parity generators 8 and 11. The amplifiers send the data read in the ROM to the data bus, and the parity generator calculates the parity value of said data, then sends the value to the comparitor 12 through the PARITY line 2. The microprocessor reads the data through the two-way amplifiers 10. The parity generator 11, activated by the line 15, calculates the parity value of the data read by the microprocessor and sends it to the comparator through the line 16. The comparator 12 compares the two values: if they are equal, the output is at the level "1"; if they are different, the output is at "0". In the latter case, the data has been altered, very likely due to a fraudulent emission of radiation. The output of the comparator is connected by the line 17 to the input "NMI" of the microprocessor. A "0" level triggers a non-maskable interrupt, which results in a redirection of the current program to an interrupt routine. The activation of the comparator can take place during the fetch phase; thus the generation of the interrupt is perfectly synchronized with the sequencing clock of the microprocessor.

The operation is identical when the microprocessor performs a read of the EEPROM.

In a simplified embodiment of the invention, the input "NMI" is similar to the RESET input of the microprocessor. When the input becomes inactive again, the microprocessor is enabled and starts its program in the same way as during a power-up.

Thus, an external intervention cannot control the microprocessor which, as long as it is subjected to this radiation, is disabled.

Compared to the first solution, which seeks to verify that an operation code is correctly read, this second solution offers the advantage of controlling any data type: operation code or data. Parity generators can also be installed in the address bus in the same way as in the data bus. This second solution also offers the advantage of controlling the signals of the address bus.

In a variant, a level "0" in the "NMI" input triggers the execution of a microprogram. In order to prevent transient phenomena, the recognition of the level of the signal in the "NMI" input takes place during a transition of the clock of the microprocessor, a transition during which the data analyzed by the parity signal comparator is valid. The operation of a conventional interrupt handling program written into the ROM would be disturbed by the hacker. According to the invention, on the other hand, a microprogram is resident in the microprocessor; it does not read data through the bus; therefore its execution cannot be disturbed by radiation acting on the bus. The microprogram, triggered by the application of a level "0" to the "NMI" input, produces two actions: the first is the writing of a flag called "DISABLED" into the programmable nonvolatile memory, and the second is a resetting of the microprocessor. The writing of the flag DISABLED is irreversible: the state of this flag can no longer be modified, either by the microprocessor or by an external means. An initialization circuit, activated when the security module is powered up, tests the state of the flag DISABLED and sets the microprocessor to a permanent RESET state if the flag has been written. Advantageously, this flag can be embodied by a fuse, the melting of which renders the microprocessor unusable.

If a hacker imposes a level "0" on all the lines of the bus, including the PARITY line, the comparator 12 does not detect any error. In essence, the number of lines at "0" being even, the parity signal must be at "0"; the PARITY line is forced to exactly that level. In order to prevent this, according to a variant of the second solution, one uses the fact that parity generators generally have two outputs, one for generating an even parity, the other for generating an odd parity.

Fig. 5 shows the wiring diagram of the modifications to be made to Fig. 4 in order to integrate programmable parity generator circuits into the monolithic circuit. The elements common to Fig. 4 and Fig. 5 have the same references.

The parity generators 7a, 8a and 11a have two outputs: the one marked Qp is the output for the even parity signal, and the other Qi is the output for the odd parity signal. When the number of input lines at "1" is even, then Qp is at "1" and Qi is at "0"; when the number of input lines at "1" is odd, then Qi is at "1" and Qp is at "0". The circuits 20 and 21, each constituted by two AND gates, one OR gate and one inverter gate, constitute multiplexers. The two inputs of the multiplexers are respectively connected to the two outputs Qi and Qp of the parity generators. A control line marked 23 selects the input. If the line 23 is at "0", the outputs of the gates 20c and 21c are at "0", and with the help of the inverter gates 20b and 20c, the outputs of the gates 20a and 21a as well as those of the gates 20d and 21d reproduce the logical levels of the outputs Qi of the parity generators 7a, 11a. In this case, the outputs Qi are selected and the data present in the outputs Qi are sent to the comparator 12. It is the odd parity signals that are compared. If the line 23 is at "1", with the help of the inverter gates 20b and 20c, the outputs of the gates 20a and 21a are at "0", the outputs of the gates 20c and 21c, as well as those of the gates 20d and 21d, reproduce the logical levels of the outputs Qp of the parity generators 7a and 11a. In the latter case, the outputs Qp are selected and the data present in the outputs Qp are sent to the comparator 12. It is the even parity signals that are compared.

The signals of the line 23 are sent by a random signal generator 22. This generator is an electronic circuit that receives through an input line the clock of the microprocessor and delivers a signal "0" and "1" randomly over time. In a simplified way, the random signal generator can be a counter whose output changes its state at each predetermined time interval. The random signal generator 22, the multiplexer 21 and the comparator 12 are located as close as possible to the microprocessor, and preferably integrated into the latter. Thus, they are not very disturbed by a hacker sending radiation through the bus. More complicated circuits using an oscillator inside the random signal generator can be installed. The purpose of this generator is to emit a logical signal through a line whose state changes fairly often, on the order of 100 to 10,000 times per second. It is important to synchronize the line's changes of state with the clock of the microprocessor, in order to

avoid accepting the "NMI" line at the precise moment when the signal of line 23 changes; otherwise interference could be produced due to differences in the propagation of the signals.

During use, the hacker subjects the data bus to radiation, which forces the data and parity bits to "1" or to "0". Depending on the binary value, the parity value calculated from the data bits forced by the radiation can be equal to the value forced by the radiation; hence, since they are equal, no error is detected. By frequently changing the parity type by means of the random signal generator, the presence of the radiation is detected immediately.

When the random signal generator 22 emits a level "0", the signals emitted by the outputs Qi of the parity generators 7a and 11a are selected and hence, the parity signals of the odd type are compared. When the random signal generator 22 emits a level "1", the signals emitted by the outputs Qp of the parity generators 7a and 11a are selected and hence, the parity signals of the even type are compared.

Let us distinguish, first of all, the case where the bus is subjected to a radiation that forces the lines of the data and parity busses 2a to "1". When the random signal generator 22 sends a signal "0", the output Qi of the parity generator 11 is selected, and its level, equal to "0", is different from that of the parity line 2a, which is forced to "1". The comparator therefore easily detects this type of radiation, triggering an interrupt.

Let's go to the second case, where the bus is subjected to a radiation that forces the lines of the data and parity busses 2a to "0". When the random signal generator 22 sends a signal "1", the output Qp of the parity generator 11 is selected, and its level, equal to "1", is different from that of the parity line 2a, which is forced to "0". In this case, too, the comparator detects the radiation, and signals it to the microprocessor through an interrupt.

This variant adds a paramater that makes the behavor of the circuit even more unpredictable for a hacker, since this parameter increases the difficulty of controlling the state of the bus lines from the outside without the circuit's detecting it.

A third solution for detecting an alteration in the execution of a program is to implement flag modification routines from place to place within a program to be

protected, and to verify prior to starting an operation on the sensitive data that all the flags have been modified.

Fig. 6 shows a part of the memory containing a program implemented at the hexadecimal address 0800. This memory can be the ROM or the EEPROM, but any other type of nonvolatile memory capable of executing instructions will work. All the flags are represented by bits and grouped in memory into an indicator. In the example described, this indicator is a byte of the RAM called FLAG. A certain number of bits composing this byte are used to mark the passage to certain steps in the program that lead to an operation on sensitive data.

At the address 800 (step 1), the FLAG byte is set to the binary value "0000 0001", the first bit at "1" indicating that step 1 has been executed. At the address 880H 9step 2) the FLAG byte is read and modified by the execution of a logical OR (the ORA instruction in MOTOROLA 6805) between its current content and the binary value "0000 0010": the result of the OR operation is written into the FLAG byte. At the address 8A0H (step 3), the FLAG byte is read and modified by the execution of a logical OR between its content and the binary value "0000 0100": the result of the OR operation is written into the FLAG byte. Finally, at the address 900H (step 4), the protected part of the program ends: the FLAG byte is read and checked: if its value is different from the binary value "0001 0111", the program jumps to a fraud handling routine.

In use, the program to be protected begins at the address 0800H. The first step (step 1) consists of updating the FLAG byte by setting the first bit of the FLAG byte to "1". Then, the program continues in sequence up to a second, so-called modification step (step 2) in which the second bit of the FLAG byte is set, thus indicating that step 2 has been executed. Consequently, we can assume that all the instructions of the program between step 1 and step 2 have been executed. Then, the program continues in sequence up to the modification step 3, where the third bit of FLAG is set to "1". Finally, the program to be protected ends in a routine for testing the FLAG byte (step 4): it consists of verifying the execution of steps 1, 2 and 3. If the value is different from "0000 0111", a trap of the program occurs, which reveals an abnormal operation, very likely resulting from an attempted fraud. In this case, the program interrupts its normal operation to jump to the fraud handling routine. Since a hacker does not know the places in the program

where the routines for modifying the FLAG byte are implemented, he does not know the moments at which they are executed, and therefore, by interfering with the values of the data bus, there is a strong probability that he will succeed in eliminating the execution of at least one of steps 1, 2 or 3, and hence the FLAG byte will not have the expected final value.

In a simplified way, the fraud handling routine can consist in a reset of the microprocessor (RESET). An improvement consists of using a microprogrammed instruction for step 4 so that a hacker cannot interfere with its flow by preventing the execution of certain instructions while allowing others. This instruction has the following structure: "Operation code, Address to be read, Value to be compared": it sequentially executes the following functions:

1 reading the "Address" byte
2 comparing the value of the byte read to "Value"
3 if equal, then jumping to the next instruction
otherwise 4 writing the flag DISABLED into nonvolatile memory
5 RESET of the microprocessor

Quite clearly, by increasing the number of flags and hence the number of steps for updating the indicator, one increases the moments at which external interference is detected. The FLAG indicator must therefore be represented by several bytes. But the steps for updating the indicator take up memory space that cannot be used for the application program, both at the level of the program and at the level of the RAM. It is therefore necessary to optimise the number of steps. For example, for a 1000-byte program to be protected, a good compromise would be to install 32 steps for modifying the indicator. These 32 steps and the final test routine take up 162 bytes of program memory and 4 bytes of RAM. Depending on the complexity of the program, which may include jumps and may not perform all of the modification steps, the test of the FLAG byte may involve only a limited number of bits. If the program to be protected ends at different places, several test routines that include different values of the FLAG byte can be installed in the program.

This solution has the advantage of being easy to use in a conventional component because it does not involve modifying the hardware portion of the component.

An improvement of the invention consists of implementing in the program instructions for erasing the FLAG byte in places that are normally never reached during the execution of the program. Thus, a fraudulent interference with the execution of the program can result in the execution of one of these instructions, which, by setting the FLAG byte to 00, results in the execution of the fraud handling routine.

Fig. 7 shows the appearance of the program memory according to the improvement. At the address 0890H (step 2a), the program executes an unconditional jump instruction. The instruction at the next address is therefore never executed, unless another jump instruction specifies it as a destination. The programmer writes an instruction for erasing the FLAG byte just after the unconditional jump instruction, and is careful never to specifiy it as a destination in his program. If a hacker disturbs the data bus, there is a certain level of probability that an unconditional jump instruction will not be executed and that this erase instruction will. This instruction sets all the bits of the FLAG byte to "0". During the final test in step 5, bits 1 and 2 of FLAG are at "0", and hence the value read is not the one expected. The program therefore interrupts its normal operation to jump to the fraud handling routine.

The program illustrated by Fig. 7 shows two steps for erasing the FLAG byte (steps 2a and 3a). The erase instruction takes up only two bytes in program memory, as opposed to four bytes for the steps for updating the indicator, which at an equal level of performance saves space. An optimal program uses fairly few unconditional jump instructions. It is therefore possible to systematically place an erase instruction after an uncondtional jump.

CLAIMS

1. Method for protecting the processing of sensitive information in a security module with a monolithic structure, the module comprising information processing means (31) and means for storing (32,33) information capable of being processed by said processing means, characterized in that it comprises the following steps:

- selecting a piece of sensitive information in the storage means;
- determining a specific condition for the integrity of said information;
- reading the information and transmitting it to the processing means;
- verifying during the processing of the information that the specific condition is satisfied;
- disabling the processing of the information if the specific condition is not satisfied.

2. Method according to claim 1, wherein the information is an operation code read in the storage means (32, 33), all of the operation codes being contained in a table having a content determined during the manufacture of the security module, and the specific integrity condition is the fact that the value of the information is equal to one of several set values.

3. Method according to claim 2, wherein the operation code to be processed is coded in the form of data bits and said bits do not all have the same binary value.

4. Method according to claim 1, wherein the specific integrity condition consists of checking a calculated piece of integrity data using the information read in the storage means (32, 33), the integrity data being calculated during the reading of the information and being transmitted to the processing means, the processing means calculating another piece of integrity data from the information received and checking for equality between the two integrity data.

5. Method according to claim 4, wherein the integrity data are calculated from at least one piece of calculation data whose value varies as a function of time.

6. Method according to claim 4, wherein the integrity data are calculated from at least one piece of calculation data whose value varies randomly.

7. Method according to claim 1, wherein the disabling of the processing of the information is performed by a microprogrammed instruction.

8. Method according to claim 7, wherein the microprogrammed instruction performs the following steps:

- writing a piece of disable data into a nonvolatile location of the storage means (32, 33);
- disabling the processing of the information.

9. Method according to claim 8 wherein, at the power-up of the module, a nonvolatile location of the storage means (32, 33) is read by the processing means (31), and the module is disabled if a value read at this location does not match.

10. Security module constituted by an electronic circuit with a monolithic structure and comprising information processing means (31) and information storage means (32, 33), the processing means selecting information extracted from the storage means in order to process it, characterized in that the processing means comprise means for verifying a specific integrity condition of a piece of sensitive information, and means for disabling the processing of the information, said disabling means being activated when the verification means have detected that the specific condition is not satisfied.

11. Security module according to claim 10, wherein the processing means (31) execute instructions corresponding to operation codes extracted from a table, characterized in that the table comprises a forbidden instruction value.

12. Security module according to claim 11, wherein the operation code to be processed is coded in the form of data bits, the security module comprising a means for reading the values of all the bits and a disabling means activated when the values of the bits are all identical.

13. Security module according to claim 10, wherein the processing means (31) execute instructions corresponding to operation codes extracted from a table, the security module comprising a means for reading an operation code and a disabling means activated during the reading of a forbidden operation code.

14. Security module according to claim 13, wherein the disabling means comprise a means for irreversibly writing an indicator into the storage means (32, 33), and a means for reading said indicator during the next power-up of the module.

15. Security module according to claim 10, comprising parity generators (7, 8) cooperating with the storage means, parity generators (11) cooperating with the processing means, and a comparator connected to each of the parity generators and capable of inducing an interrupt in the processing means.

16. Security module according to claim 15, wherein the operation of the parity generators (7, 8) varies as a function of time.

17. Security module according to claim 15, wherein the operation of the parity generators (7, 8) varies randomly.

18 Security module according to claim 14, characterized in that the irreversible writing of the indicator into the storage means (32, 33) is performed by executing a microprogrammed instruction.

19. Security module according to claim 10, characterized in that the security module is a microcircuit card.

ABSTRACT

METHOD FOR MAKING SECURE A SENSITVE INFORMATION PROCESSING IN A MONOLITHIC SECURITY MODULE, AND ASSOCIATED SECURITY MODULE

The invention relates to a method for protecting the processing of sensitive information in a security module with a monolithic structure, the module comprising information processing means (9) and means for storing (3, 4) information capable of being processed by said processing means, characterized in that it comprises the following steps:

- selecting a piece of sensitive information in the storage means;
- determining (7) a specific condition for the integrity of said information;
- reading the information and transmitting (1) it to the processing means;
- verifying (11) during the processing of the information that the specific condition is satisfied;
- disabling the processing of the information if the specific condition is not satisfied.

The invention also relates to the associated security module.

Fig. 4

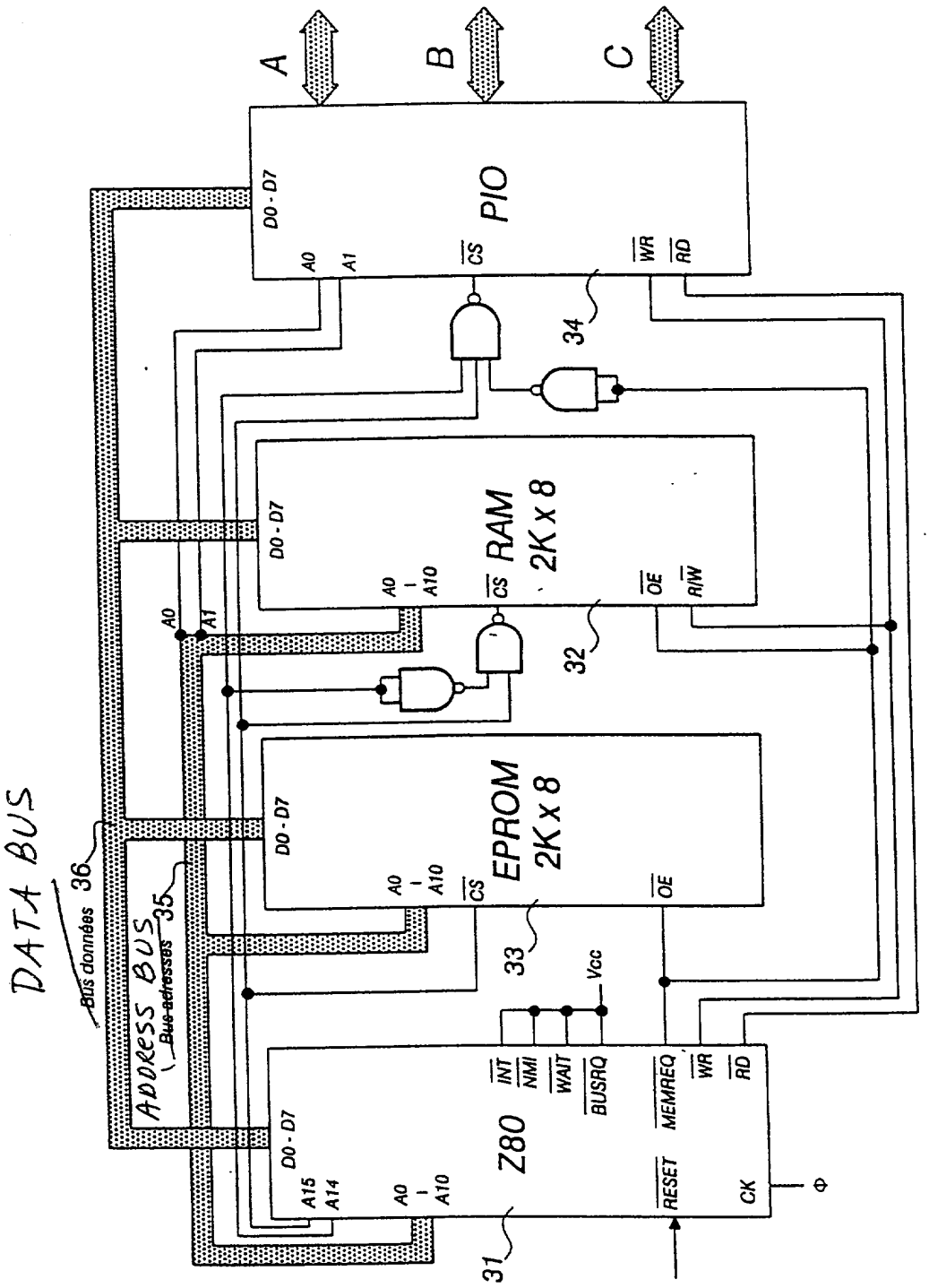
A
B
C
PIO
D0 - D7
A0
A1
CS
WR
RD
34
RAM 2K x 8
A0 - A10
OE
R/W
32
EPROM 2K x 8
33
Vcc
Z80
INT
NMI
WAIT
BUSRQ
MEMREQ
RESET
CK
Φ
A15
A14
31
DATA BUS
Bus données
36
ADDRESS BUS
Bus adresses
35

Fig.1

HIGH-ORDER

\Poids forts / LOW-ORDER \Poids faibles	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0	0000	BRSET0	BSET0	BRA	NEG	NEG	NEG	NEG	NEG	RTI		SUB	SUB	SUB	SUB	SUB	SUB
1	0001	BRSET1	BSET1	BRN						RTS		CMP	CMP	CMP	CMP	CMP	CMP
2	0010	BRSET2	BSET2	BHI								SBC	SBC	SBC	SBC	SBC	SBC
3	0011	BRSET3	BSET3	BLS	COM	COM	COM	COM	COM	SWI		CPX	CPX	CPX	CPX	CPX	CPX
4	0101	BRSET4	BSET4	BCC	LSR	LSR	LSR	LSR	LSR			AND	AND	AND	AND	AND	AND
5	0101	BRSET5	BSET5	BCS								BIT	BIT	BIT	BIT	BIT	BIT
6	0110	BRSET6	BSET6	BNE	ROR	ROR	ROR	ROR	ROR			LDA	LDA	LDA	LDA	LDA	LDA
7	0111	BRSET7	BSET7	BEQ	ASR	ASR	ASR	ASR	ASR		TAX		STA	STA	STA	STA	STA
8	1000	BRCLR0	BCLR0	BHCC	LSL	LSL	LSL	LSL	LSL		CLC	EOR	EOR	EOR	EOR	EOR	EOR
9	1001	BRCLR1	BCLR1	BHCS	ROL	ROL	ROL	ROL	ROL		SEC	ADC	ADC	ADC	ADC	ADC	ADC
A	1010	BRCLR2	BCLR2	BPL	DEC	DEC	DEC	DEC	DEC		CLI	ORA	ORA	ORA	ORA	ORA	ORA
B	1011	BRCLR3	BCLR3	BMI							SEI	ADD	ADD	ADD	ADD	ADD	ADD
C	1100	BRCLR4	BCLR4	BMC	INC	INC	INC	INC	INC		RSP		JMP	JMP	JMP	JMP	JMP
D	1101	BRCLR5	BCLR5	BMS	TST	TST	TST	TST	TST		NOP	BSR	JSR	JSR	JSR	JSR	JSR
E	1110	BRCLR6	BCLR6	BIL								LDX	LDX	LDX	LDX	LDX	LDX
F	1111	BRCLR7	BCLR7	BIH	CLR	CLR	CLR	CLR	CLR	WAIT	TXA		STX	STX	STX	STX	STX

Fig.2

HIGH-ORDER

\Poids forts / \Poids faibles LOW-ORDER	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	FRD	RTI	BSET0	BSET0	BRA	NEG	NEG	NEG	NEG	NEG	SUB	SUB	SUB	SUB	SUB	SUB
1 0001	FRD	RTS	BSET1	BSET1	BRN	FRD	FRD	FRD	FRD	FRD	CMP	CMP	CMP	CMP	CMP	CMP
2 0010	FRD	FRD	BSET2	BSET2	BHI	FRD	FRD	FRD	FRD	FRD	SBC	SBC	SBC	SBC	SBC	SBC
3 0011	FRD	SWI	BSET3	BSET3	BLS	COM	COM	COM	COM	COM	CPX	CPX	CPX	CPX	CPX	CPX
4 0101	FRD	FRD	BSET4	BSET4	BCC	LSR	LSR	LSR	LSR	LSR	AND	AND	AND	AND	AND	AND
5 0101	FRD	FRD	BSET5	BSET5	BCS	FRD	FRD	FRD	FRD	FRD	BIT	BIT	BIT	BIT	BIT	BIT
6 0110	FRD	FRD	BSET6	BSET6	BNE	ROR	ROR	ROR	ROR	ROR	LDA	LDA	LDA	LDA	LDA	LDA
7 0111	TAX	FRD	BSET7	BSET7	BEQ	ASR	ASR	ASR	ASR	ASR	STA	STA	STA	STA	STA	FRD
8 1000	CLC	FRD	BCLR0	BCLR0	BHCC	LSL	LSL	LSL	LSL	LSL	EOR	EOR	EOR	EOR	EOR	EOR
9 1001	SEC	FRD	BCLR1	BCLR1	BHCS	ROL	ROL	ROL	ROL	ROL	ADC	ADC	ADC	ADC	ADC	ADC
A 1010	CLI	FRD	BCLR2	BCLR2	BPL	DEC	DEC	DEC	DEC	DEC	ORA	ORA	ORA	ORA	ORA	ORA
B 1011	SEI	FRD	BCLR3	BCLR3	BMI	FRD	FRD	FRD	FRD	FRD	ADD	ADD	ADD	ADD	ADD	ADD
C 1100	RSP	FRD	BCLR4	BCLR4	BMC	INC	INC	INC	INC	INC	JMP	JMP	JMP	JMP	JMP	FRD
D 1101	NOP	FRD	BCLR5	BCLR5	BMS	TST	TST	TST	TST	TST	JSR	JSR	JSR	JSR	JSR	BSR
E 1110	FRD	FRD	BCLR6	BCLR6	BIL	FRD	FRD	FRD	FRD	FRD	LDX	LDX	LDX	LDX	LDX	LDX
F 1111	TXA	WAIT	BCLR7	BCLR7	BIH	CLR	CLR	CLR	CLR	CLR	STX	STX	STX	STX	STX	FRD

Fig.3

09/763868

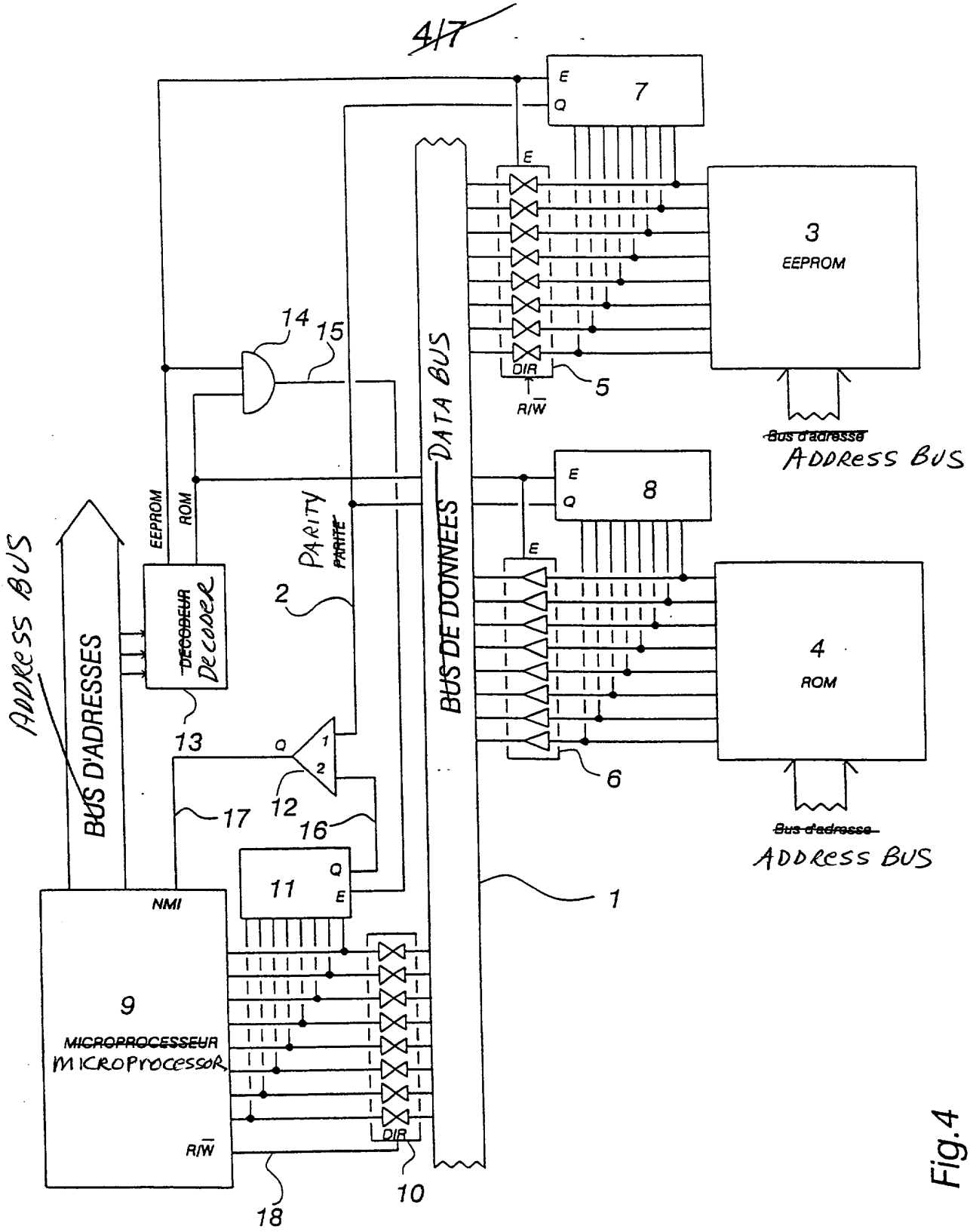
E
Q
7
E
3
EEPROM
DIR
R/W
5
Bus d'adresse
ADDRESS BUS
14
15
DATA BUS
BUS DE DONNEES
E
Q
8
E
4
ROM
6
Bus d'adresse
ADDRESS BUS
EEPROM
ROM
DECODEUR
Decoder
PARITY
PARITE
2
13
Q
1
2
12
17
16
11
Q
E
1
NMI
9
MICROPROCESSEUR
MICROPROCESSOR
R/W
DIR
10
18
ADDRESS BUS
BUS D'ADRESSES

Fig.4

5/7

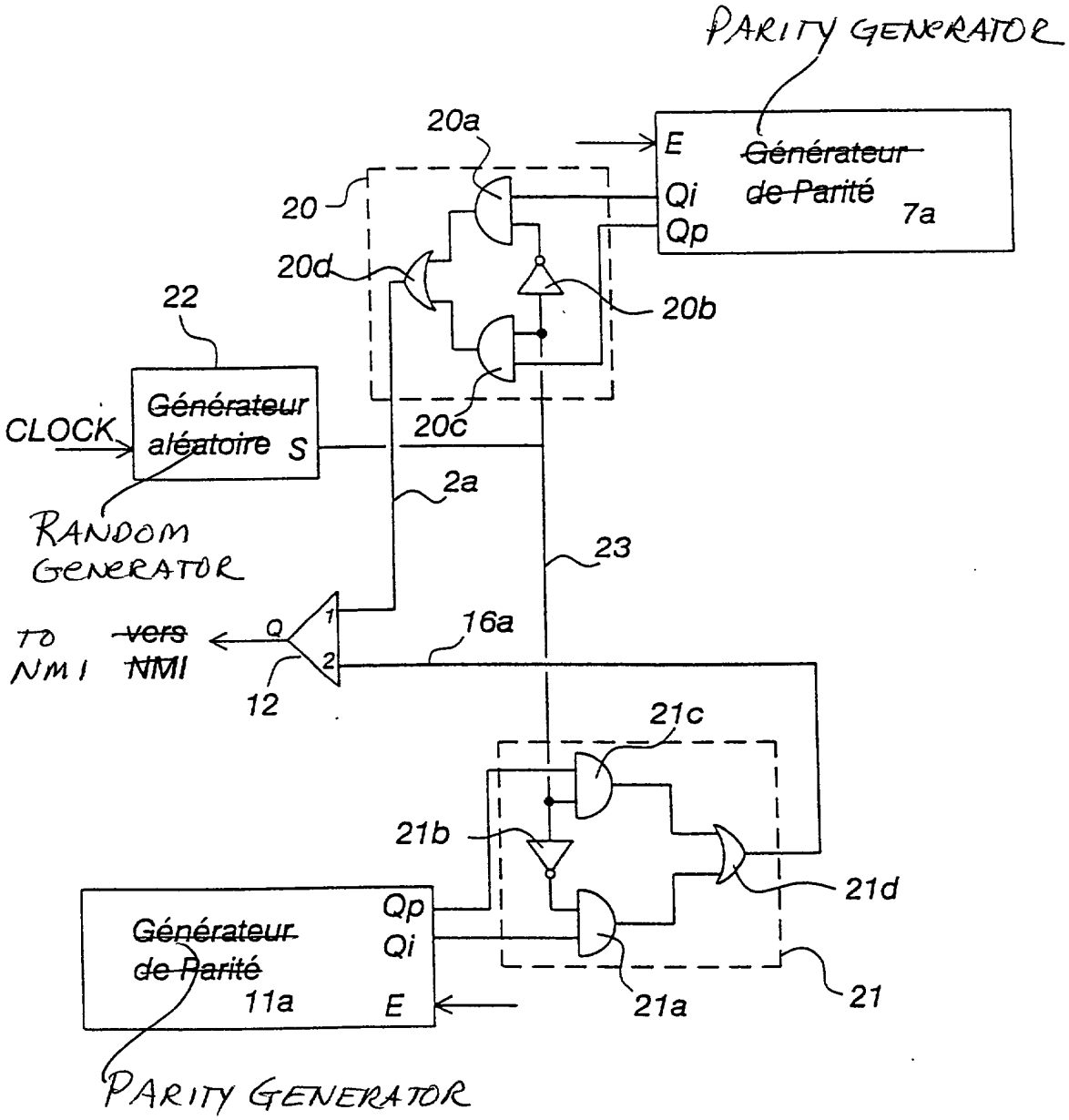
PARITY GENERATOR
E
Générateur
de Parité
7a
Qi
Qp
20a
20
20d
20b
22
Générateur
aléatoire
S
CLOCK
20c
2a
RANDOM
GENERATOR
23
TO
NMI
vers
NMI
Q
1
2
16a
12
21c
21b
21d
Qp
Qi
E
Générateur
de Parité
11a
21a
21
PARITY GENERATOR

Fig.5

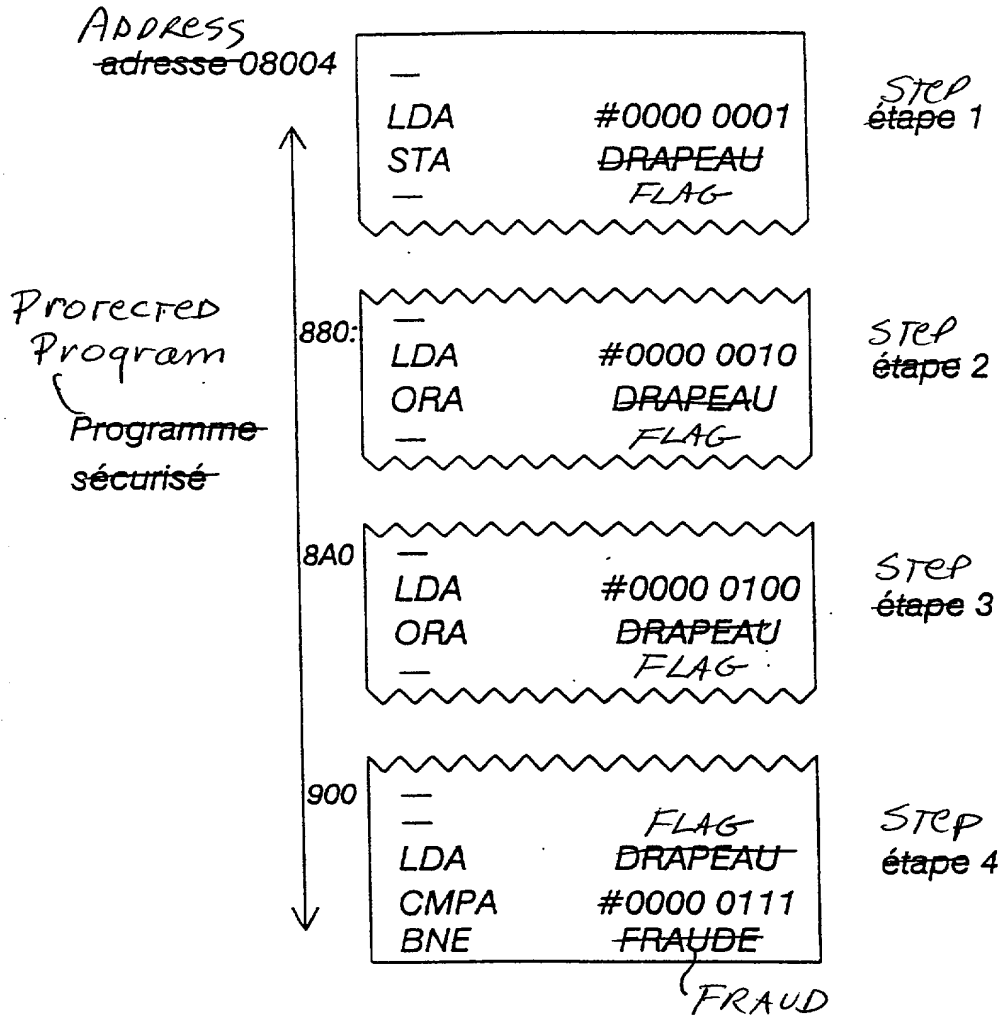
ADDRESS
~~adresse~~ 08004
Protected Program
~~Programme sécurisé~~
—
LDA #0000 0001
STA ~~DRAPEAU~~ FLAG
—
STEP ~~étape~~ 1
880:
—
LDA #0000 0010
ORA ~~DRAPEAU~~ FLAG
—
STEP ~~étape~~ 2
8A0
—
LDA #0000 0100
ORA ~~DRAPEAU~~ FLAG
—
STEP ~~étape~~ 3
900
—
—
LDA FLAG ~~DRAPEAU~~
CMPA #0000 0111
BNE ~~FRAUDE~~ FRAUD
STEP ~~étape~~ 4

Fig.6

7/7

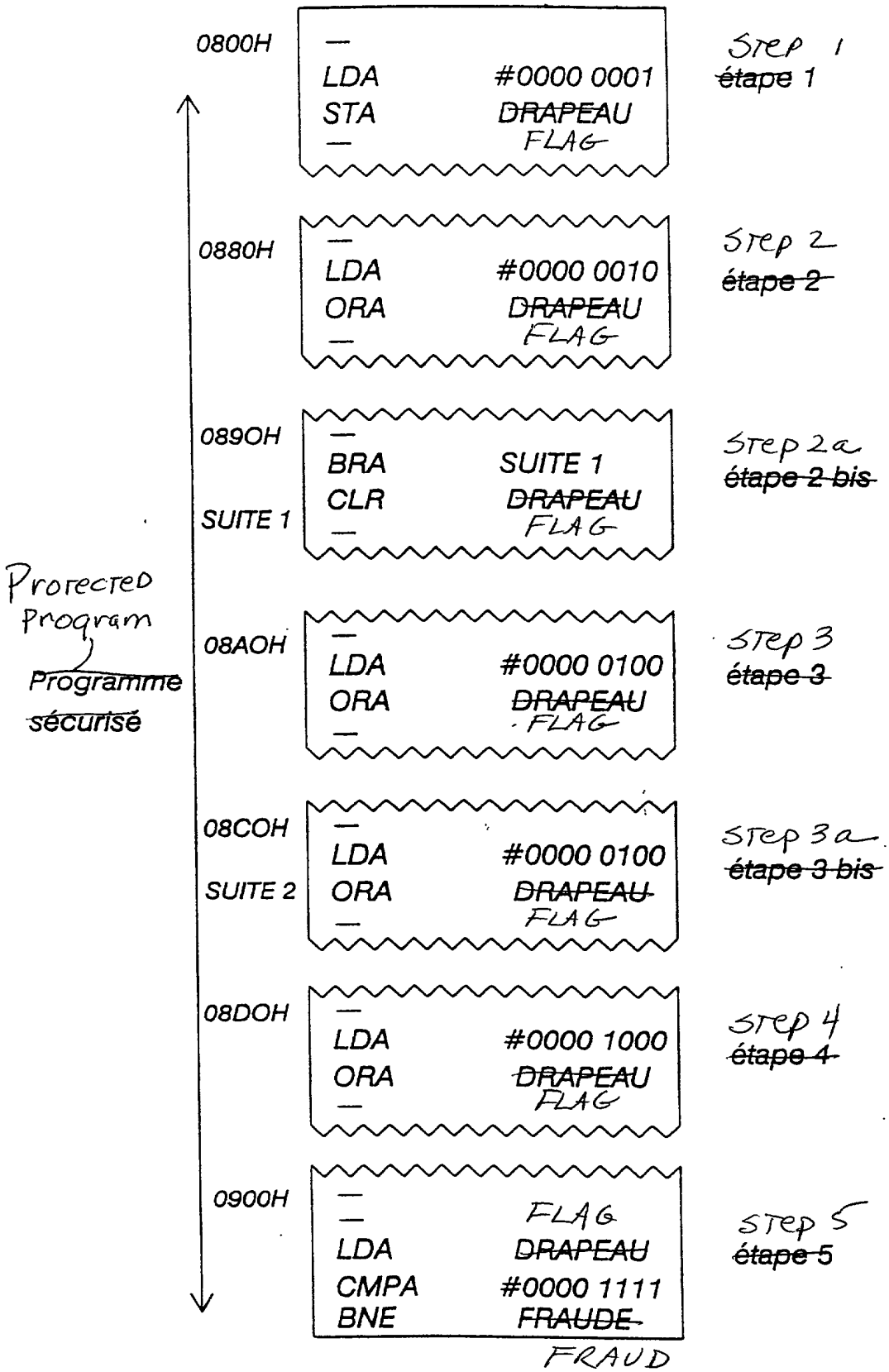
0800H
—
LDA #0000 0001
STA DRAPEAU
— FLAG
Step 1
étape 1
0880H
—
LDA #0000 0010
ORA DRAPEAU
— FLAG
Step 2
étape 2
089OH
—
BRA SUITE 1
SUITE 1 CLR DRAPEAU
— FLAG
Step 2a
étape 2 bis
Protected Program
Programme sécurisé
08AOH
—
LDA #0000 0100
ORA DRAPEAU
— FLAG
Step 3
étape 3
08COH
—
LDA #0000 0100
SUITE 2 ORA DRAPEAU
— FLAG
Step 3a
étape 3 bis
08DOH
—
LDA #0000 1000
ORA DRAPEAU
— FLAG
Step 4
étape 4
0900H
—
— FLAG
LDA DRAPEAU
CMPA #0000 1111
BNE FRAUDE
FRAUD
Step 5
étape 5

Fig.7

Declaration and Power of Attorney For Patent Application
Declaration Pour Demandes de Brevets Avec Pouvoirs

French Language Declaration

En tant qu' inventeur nomme ci-après, Je déclare par le présent acte que:

Mon nom, mon domicile, mon adresse postale, ma nationalité sont ceux qui figurent ci-apres,

Je déclare que je crois être l'inventeur original, premier et unique (si un seul nom figure sur le présent acte) ou un des co-inventeurs, originaux et premiers (si plusieurs noms figurent sur le present acte) du sujet revendiqué et pour liquel un brevet est demande sur la base de l'invention intitulée:

~~Procédé de sécurisation du traitement d'une~~ **information sensible dans un module de** ~~sécurité monolithique, et module de sécurité~~ **associé.**

dont la description
(cocher la case correspondante)

☒ est annexée au présent acte.

☐ a été déposée _____

Numéro de série de la demande _____

et modifiée le _____
(si approprié)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

_____

_____

_____,

the specification of which

(check one)

☐ is attached hereto.

☐ was filed on _____ as

Application Serial No. _____

and was amended on _____.
(if applicable)

Je déclare par le présent acte avoir examiné et compris le contenu de la description identifiee ci-dessus, revendications y compris, et le cas échéant telle que modifiée par l'amendment cité plus haut.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

Je reconnais le devoir de divulguer l'information qui est en rapport avec l'examen de cette demande selon Titre 37 du Code des Reglements Fédéraux §1.56(a).

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

French Language Declaration

Je revendique par le present acte le bénéfice de priorite étrangère selon Titre 35, du Code des Etats-Unis, §119 de toute demande de brevet ou d'attestation d'inventeur énumérée ci-apres, et j'ai identifie egalement ci-apres toute demande etrangere de brevet ou d'attestation d'inventeur ayant une date de dépôt anterieure à celle de la demande pour laquelle la priorite est revendiquee.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior foreign applications Demande(s) de brevet anterieure(s) dans un autre pays:			Priority claimed Droit de priorite revendique	
FR 99 08409 (Number) (Numéro)	France (Country) (Pays)	30 06 1999 (Day/Month/Year Filed) (Jour/Mois/Annee de depôt)	X Yes Oui	No Non
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Annee de depôt)	Yes Oui	No Non
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Annee de depôt)	Yes Oui	No Non

Je revendique par le présent acte, le bénéfice selon Titre 35 du Code des Etats-Unis, §120 de toute(s) demande(s) américaines énumérée(s) ci-après et, dans la mesure où le sujet de chacune des revendications de cette demande n'est pas divulgué dans la demande américaine anterieure, de la façon définie par le premier paragraphe de Titre 35 du Code des Etats-Unis, §112, je reconnais le devoir de divulguer l'information pertinente selon Titre 37 du Code des Reglements Fédéraux, §1.56(a), toute information qui se présente entre la date de dépôt de la demande anterieure et la date de dépôt de la demande, soit nationale, soit internationale PCT.

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.) (No. de Demande)	(Filing Date) (Date de Depôt)	(Etat) (brevetée, pendante, abandonne)	(Status) (patented, pending, abandoned)
PCT/FR00/01814	29/06/2000	PENDING	

Je déclare par le present acte que toutes mes déclarations, à ma connaissance, sont vraies et que toutes les déclarations faites à partir de renseignements ou de suppositions, sont tenues pour être vraies; de plus, toutes ces declarations ont été faites en sachant que de fausses déclarations volontaires u autres actes de même nature sont sanctionées par une amende ou un emprisonnement, ou les deux, selon la Section 1001, du Titre 18 de Code des Etats-Unis et que de selles déclarations délibérément fausses peuvent compromettre la validité de la demande ou du brevet délivré.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

French Language Declaration

POUVOIR: En tant qu'inventeur, je designe l'(les) avocat(s) et/ou l'(les) agent(s) suivant(s) pour poursuivre la procédure de cette demande et traiter toute affaire la concernant supris du Bureau des Brevets et de Marques:

Harold L. Stowell, Reg. 17,233
Edward J. Kondracki, Reg. 20,604
Dennis P. Clarke, Reg. 22,549
William L. Feeney, Reg. 29,918
John C. Kerins, Reg. 32,421

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (*list name and registration number*)

Harold L. Stowell, Reg. 17,233
Edward J. Kondracki, Reg. 20,604
Dennis P. Clarke, Reg. 22,549
William L. Feeney, Reg. 29,918
John C. Kerins, Reg. 32,421

Adresser toure correspondance à:

Edward J. Kondracki, Esq.
KERKAM, STOWELL, KONDRACKI
& CLARKE, P.C.
5203 Leesburg Pike, Suite 600
Falls Church, VA 22041

Send Correspondence to:

Edward J. Kondracki, Esq.
KERKAM, STOWELL, KONDRACKI
& CLARKE, P.C.
5203 Leesburg Pike, Suite 600
Falls Church, VA 22041

Adresser toute communication téléphonique à: (*Nom*) (*Numéro de téléphone*)

Edward J. Kondracki, Esq.
(703) 998-3302

Direct Telephone Calls to: (*name and telephone number*)

Edward J. Kondracki, Esq.
(703) 998-3302

Nom complet du seul ou premier inventeur HAZARD Michel	Full name of sole or first inventor
Signature de l'inventeur Date 3/9/99	Inventor's signature Date
Domicile 27 rue des Harias 78124 MAREIL SUR MAULDRE FRANCE	Residence
Nationalité Française	Citizenship
Adresse Postale 27 rue des Harias 78124 MAREIL SUR MAULDRE FRANCE	Post Office Address
Nom complet du second co-inventeur, le cas echeant	Full name of second joint inventor, if any
Signature de l'inventeur Date	Second inventor's signature Date
Domicile	Residence
Nationalité	Citizenship
Adresse Postale	Post Office Address

(Fournir les mêmes renseignements et la signature de tout co-inventeur supplémentaire.)

(Supply similar information and signature for third and subsequent joint inventors.)